

**EV355228440**

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

**Method and Apparatus for Playback of Audio Files**

Inventors:  
Phillip Lu  
Kipley J. Olson

ATTORNEY DOCKET NO. MS1-1579US

## **RELATED APPLICATIONS**

The present application is a Continuation In Part of Serial No. 10/606,464, filed June 26, 2003, entitled "Method and Apparatus for Audio Normalization", the disclosure of which is incorporated by reference herein.

## **TECHNICAL FIELD**

The systems and methods described herein relate to fading-in one audio file while fading-out another audio file.

## **BACKGROUND**

Computer systems are being used today to store various types of media, such as audio data, video data, combined audio and video data, and streaming media from online sources. A user of a computer system can play audio data through the computer system using, for example, a media player application. Multiple audio files (e.g., songs or music tracks) may be played sequentially. As playback of a first audio file nears the end of the file, the audio file fades out (e.g., the playback volume of the first audio file gradually decreases to silence). As the first audio file fades out, playback of a second audio file (i.e., the next audio data file in the sequence) fades-in. During fade-in, the playback volume of the second audio file gradually increases from silence to the audio file's full volume.

Different audio files have varying amounts of silence at the beginning and ending of the audio files. These periods of silence may vary from less than a second to many seconds in length. Such variable periods of silence from one audio file to the next create problems when attempting to fade-in and fade-out audio files in the manner discussed above. For example, if an audio file fades-in

1 over a period of five seconds and the audio file has six seconds of silence at the  
2 beginning of the file, the entire fade-in period will be silent. In another example,  
3 if an audio file fades-out over a period of five seconds and the audio file has four  
4 seconds of silence at the end of the file, the fade-out period is effectively reduced  
5 to one second (since the remaining four seconds are silent). Thus, these variable  
6 periods of silence may interfere with fade-in and/or fade-out of audio files.  
7 Inconsistent fade-in and/or fade-out of audio files can be annoying to a listener  
8 and disturbs their listening enjoyment.

9 Accordingly, it is desirable to provide an audio playback mechanism that  
10 fades-in and/or fades-out non-silent portions of the audio files.

## 11 12 **SUMMARY**

13 The systems and methods described herein provide for the fading-in of one  
14 audio file while fading-out another audio file. In a particular embodiment, a  
15 request is received to play a first audio file and a second audio file. The systems  
16 and methods identify a first effective start position and a fade-out position  
17 associated with the first audio file. The systems and methods also identify a  
18 second effective start position associated with the second audio file. The first  
19 audio file is played from the first effective start position. When playback of the  
20 first audio file reaches the fade-out position associated with the first audio file,  
21 playback of the first audio file is faded-out while the second audio file is played  
22 from the second effective start position.

## **BRIEF DESCRIPTION OF THE DRAWINGS**

Similar reference numbers are used throughout the figures to reference like components and/or features.

Fig. 1 is a block diagram illustrating an example of various components that can be used to fade-in and fade-out playback of audio files.

Fig. 2 is a flow diagram illustrating an embodiment of a procedure for determining an effective start position of an audio file.

Fig. 3 is a flow diagram illustrating an embodiment of a procedure for determining an effective end position of an audio file.

Figs. 4A and 4B are flow diagrams illustrating an embodiment of a procedure for fading-in and fading-out audio files.

Fig. 5 is a graph illustrating an effective start position and an effective end position associated with a particular audio file.

Fig. 6 is a graph illustrating the manner in which a first audio file fades-out and a second audio file fades-in.

Fig. 7 illustrates a general computer environment, which can be used to implement the techniques described herein.

## **DETAILED DESCRIPTION**

The systems and methods discussed herein relate to fading-in one audio file while fading-out another audio file. The fading-in of one audio file while fading-out another audio file may also be referred to as “blending” or “cross-fading” two audio files. The systems and methods described identify effective start positions and effective end positions for each audio file. The effective start positions and effective end positions reduce or eliminate playback of silent portions during the

1 fade-in and fade-out of each audio file. These systems and methods provide a  
2 more consistent and uniform fade-in and fade-out of audio files, thereby enhancing  
3 the listening experience.

4 The systems and methods described herein can be applied during, for  
5 example, media playback, media recording (e.g., “burning” a CD or DVD) and  
6 media scanning or analysis. In a specific embodiment, a fade-in and/or a fade-out  
7 process is applied across multiple audio files during playback. As discussed  
8 herein, various fade-in and fade-out parameters can be calculated at different  
9 times, such as during playback of an audio file, when copying an audio file, when  
10 scanning a media library, or during media recording.

11 As used herein, the term “media clip” describes any sequence of audio data,  
12 video data, combined audio and video data, etc. A “media clip” may also be  
13 referred to as an “audio clip”, a “video clip”, or a “song”. As used herein, the term  
14 “audio file” describes any sequence of audio data having any length. An “audio  
15 file” may contain other information in addition to audio data, such as  
16 configuration information, associated video data, and the like. An “audio file”  
17 may also be referred to as a “media file”.

18 Although particular examples discussed herein refer to playing or copying  
19 audio data from CDs, the systems and methods described herein can be applied to  
20 any audio data obtained from any source, such as CDs, DVDs (digital video disks  
21 or digital versatile disks), video tapes, audio tapes and various online sources. The  
22 audio data processed by the systems and methods discussed herein may be stored  
23 in any format, such as a raw audio data format or a compressed format such as  
24 WMA (Windows Media Audio), MP3 (MPEG, audio layer 3), WAV (a format for  
25

1 storing sound in files; uses “.wav” filename extension), WMV (Windows Media  
2 Video), or ASF (Advanced Streaming Format).

3 Fig. 1 is a block diagram illustrating an example of various components  
4 that can be used to fade-in and fade-out playback of audio files. The various  
5 components shown in Fig. 1 may be included in a media player application such as  
6 the Windows Media<sup>®</sup> Player available from Microsoft Corporation of Redmond,  
7 Washington. An audio cross-fade engine 102 is coupled to a media copy module  
8 104, a media playback module 106, a media scan module 108, a media burning  
9 module 110 and a media download module 112. Audio cross-fade engine 102  
10 fades-in and fades-out volume levels for multiple audio files. Media copy module  
11 104 allows a user to copy an audio file from, for example, a CD to a computer  
12 hard drive (or other storage device) such that the audio file can be played back  
13 through the computer’s speakers. This process of copying an audio file from a CD  
14 to a computer hard drive is commonly referred to as “ripping”. Media copy  
15 module 104 may also allow a user to copy an audio file from a computer hard  
16 drive to a portable device, such as a WMA or MP3 player.

17 Media playback module 106 plays audio and/or video data from a CD,  
18 DVD, computer hard drive, or other source. Typically, media player module 106  
19 plays audio data through a computer’s speakers and plays video data on the  
20 computer’s monitor. Media scan module 108 scans storage devices coupled to a  
21 computer system for audio and/or video files and categorizes those audio and/or  
22 video files. Media scan module 108 is typically executed when a media player is  
23 installed on a new computer or when a user wants to update a listing of all audio  
24 and/or video files on the computer. Media scan module 108 generally scans hard  
25

1 drives, CD-ROM drives, DVD drives, other drives containing removable media,  
2 and any portable devices coupled to the computer.

3 Media burning module 110 controls the recording of data (such as audio  
4 and video data) on a recordable media, such as a recordable CD or a recordable  
5 DVD. The process of recording a CD or a DVD is commonly referred to as  
6 “burning” a CD or DVD. Media burning module 110 may record data from  
7 multiple sources onto a single CD or DVD. For example, a collection of audio  
8 data stored on a CD may be from another CD, an online source, and from an audio  
9 track on a DVD.

10 Media download module 112 allows users to download media content from  
11 various sources, such as web sites, music download services, or data storage  
12 mechanisms accessible via, for example, a data communication network. As  
13 media content is downloaded by media download module 112, various fade-in and  
14 fade-out parameters are computed and saved.

15 Audio cross-fade engine 102 is also coupled to a media library 114, which  
16 stores, for example, fade-in and fade-out parameters associated with multiple  
17 audio files. Additional details regarding these parameters are discussed below.  
18 Media library 114 may also contain configuration information, audio data, video  
19 data, and other data used by audio cross-fade engine 102 and the media player  
20 application.

21 Audio cross-fade engine 102 includes a cross-fade parameter calculator 116  
22 and a cross-fader 118. Cross-fade parameter calculator 116 analyzes audio data  
23 and calculates one or more fade-in and/or fade-out parameters (also referred to as  
24 “cross-fade parameters”) that are applied to the audio data during playback such  
25 that one audio file fades out as the next audio file fades in. These parameters are

1 stored in media library 114 along with an identifier of the audio data with which  
2 the parameters are associated. The cross-fade parameters are applied by cross-  
3 fader 118 during playback of the audio data to cross-fade the audio data. Cross-  
4 fader 118 may work in combination with media playback module 106 to cross-  
5 fade audio data.

6 As shown in Fig. 1, media copy module 104, media playback module 106,  
7 media scan module 108, media burning module 110 and media download module  
8 112 are coupled to audio cross-fade engine 102. Certain modules 104-112 may  
9 interact with cross-fade parameter calculator 116 and/or cross-fader 118 in engine  
10 102. In one embodiment, all five modules 104-112 are coupled to cross-fade  
11 parameter calculator 116, while media playback module 106 and media burning  
12 module 110 are also coupled to cross-fader 118.

13 Fig. 2 is a flow diagram illustrating an embodiment of a procedure 200 for  
14 determining an effective start position of an audio file. Initially, an audio file is  
15 identified to be analyzed (block 202). The procedure 200 then analyzes the first  
16 two data samples in the audio file (block 204). A typical audio file may contain  
17 thousands or millions of data samples when using a typical data sampling rate of  
18 44,000 samples per second. The procedure then calculates a running average of  
19 the past N data samples by calculating a RMS value (block 206). The value of "i"  
20 represents the index for a particular sample. In one embodiment, the running  
21 average is computed as:



$$T_i = \sqrt{\frac{\sum_{k=i-N+1}^i x_k^2}{N}}$$

The procedure then determines whether the running average exceeds a threshold value (block 208). The threshold value is calculated by first calculating an average value across the entire media clip (e.g., during the calculation discussed above) and using a fraction of that average value as the threshold value. In a particular embodiment, 1% of the average value is used as the threshold value. If the running average does not exceed the threshold value, the procedure selects the next data sample in the audio file (block 210) and returns to block 206 to calculate a new running average that includes the data sample selected in block 210.

When the running average exceeds the threshold value at block 208, the procedure marks the current data sample as the effective start position of the audio file (block 212). All previous data samples are marked as “silent” (block 214). This marking of data samples can be implemented, for example, by identifying at time position within the audio file at which the data samples are no longer considered “silent”. Any future processing of the audio file will treat all data samples from the beginning of the audio file to the identified time position as “silent”.

Another embodiment first filters the data samples to eliminate frequencies inaudible to humans, thereby leaving the portion of the audio signal that can be heard by human ears. The energy of the filtered audio signal is then calculated

1 and a threshold is used to determine whether particular data samples are "silent".  
2 In another embodiment, a system or method calculates or estimates the  
3 background noise level in the audio signal. When the magnitude of the data  
4 samples is less than (or equal to) the background noise level, those data samples  
5 are marked as "silent".

6 The information associated with the effective start position of the audio file  
7 is saved in the media library (block 216) and saved with the audio file (block 218).  
8 Thus, the threshold value mentioned above filters out data samples that are silent  
9 or nearly silent. Filtering these data samples eliminates or reduces silent playback  
10 during fade-in of the audio file.

11 Saving the effective start position information "with the audio file" includes  
12 editing the audio file to include the information or storing the information in  
13 another file adjacent to or associated with the audio file. In one embodiment, the  
14 effective start position information is stored in a header of the audio file. In  
15 certain situations, the audio file cannot be edited and the information cannot be  
16 stored with the audio file. In these situations, the audio cross-fade engine relies on  
17 the effective start position information stored in the media library during playback  
18 of the audio file.

19 Fig. 3 is a flow diagram illustrating an embodiment of a procedure 300 for  
20 determining an effective end position of an audio file. Initially, an audio file is  
21 identified to be analyzed (block 302). The procedure 300 then analyzes the last  
22 two data samples in the audio file (block 304). The procedure calculates a running  
23 average of the past N data samples by calculating a RMS value (block 306). In  
24 one embodiment, the running average is computed using the formula discussed  
25 above with respect to Fig. 2.

1       The procedure then determines whether the running average exceeds a  
2 threshold value (block 308). In one embodiment, the threshold value used in this  
3 determination is the same threshold value discussed above with respect to block  
4 208 (Fig. 2). If the running average does not exceed the threshold value, the  
5 procedure selects the previous data sample (i.e., moving backwards) in the audio  
6 file (block 310) and returns to block 306 to calculate a new running average that  
7 includes the data sample selected in block 310.

8       When the running average exceeds the threshold value at block 308, the  
9 procedure marks the current data sample as the effective end position of the audio  
10 file (block 312). All previous data samples (i.e., data samples between the current  
11 data sample and the end of the audio file) are marked as “silent” (block 314). The  
12 information associated with the effective end position of the audio file is saved in  
13 the media library (block 316) and saved with the audio file (block 318). Thus, the  
14 threshold value mentioned above filters out data samples that are silent or nearly  
15 silent. Filtering these data samples eliminates or reduces silent playback during  
16 fade-out of the audio file.

17       As discussed above with respect to Fig. 2, saving the effective end position  
18 information “with the audio file” includes editing the audio file to include the  
19 information or storing the information in another file adjacent to or associated with  
20 the audio file. In one embodiment, the effective end position information is stored  
21 in a header of the audio file.

22       The procedures discussed above with respect to Fig. 2 and Fig. 3 may be  
23 performed independently or along with another procedure. For example, when  
24 copying an audio file to a hard disk drive or other storage device, the procedures  
25 of Fig. 2 and Fig. 3 may be performed along with the process of copying the audio

1 file. Thus, the effective start position and the effective end position can be stored  
2 with the copied audio file for use by a media player application during playback of  
3 the copied audio file.

4 Figs. 4A and 4B are flow diagrams illustrating an embodiment of a  
5 procedure for fading-in and fading-out audio files. Initially, a user selects multiple  
6 audio files for playback (block 402). For example, a user may select multiple files  
7 individually, may select an album containing multiple songs, or may select a  
8 playlist containing multiple songs. An audio cross-fade engine identifies an  
9 effective start position and a fade-out position of a first audio file (block 404).  
10 The fade-out position of the first audio file can be determined using the effective  
11 end position of the audio file and the duration of the fade-out process. For  
12 example, if the effective end position of the first audio file is 3:26 (three minutes  
13 and 26 seconds) from the effective start position of the first audio file and the  
14 duration of the fade-out process is five seconds, the fade-out position for the first  
15 audio file is 3:21 (3:26 – 5 seconds). Thus, if the first audio file begins fading out  
16 at 3:21, it will be completely faded out at 3:26 (the effective end position of the  
17 first audio file). The examples discussed herein assume that the duration of the  
18 fade-out process is five seconds. Similarly, the examples discussed herein assume  
19 that the duration of the fade-in process is also five seconds. In alternate  
20 embodiments, the duration of the fade-in process and the duration of the fade-out  
21 process can be any time period. In particular embodiments, the duration of the  
22 fade-in process matches the duration of the fade-out process. However, in other  
23 embodiments, the duration of the fade-in process may be longer or shorter than the  
24 duration of the fade-out process.

1 In a particular embodiment, the volume level of the audio file fades-in or  
2 fades out in a linear manner. For example, if the fade-in period is five seconds, the  
3 volume fades-in from zero to the full volume linearly during the five second  
4 period. In alternate embodiments, another function, such as an exponential  
5 function, is used to fade-in and/or fade-out the volume level.

6 Referring again to Fig. 4A, procedure 400 begins fading-in playback of the  
7 first audio file (block 406). The procedure then determines whether playback of  
8 the first audio file has reached the fade-out position (block 408). If not, the  
9 procedure continues playing the first audio file (block 410) until the fade-out  
10 position is reached. When playback of the first audio file reaches the fade-out  
11 position, the procedure begins fading-out playback of the first audio file (block  
12 412).

13 Referring to Fig. 4B, as the first audio file begins fading-out, the audio  
14 cross-fade engine identifies an effective start position and a fade-out position of a  
15 second audio file (block 414). The procedure then begins fading-in playback of  
16 the second audio file (block 416). Thus, playback of the second audio file fades-in  
17 as playback of the first audio file fades-out.

18 The procedure continues by determining whether playback of the second  
19 audio file has reached the fade-out position (block 418). If not, the procedure  
20 continues playing the second audio file (block 420) until the fade-out position is  
21 reached. When playback of the second audio file reaches the fade-out position, the  
22 procedure begins fading-out playback of the second audio file (block 422). If a  
23 third audio file is to be played after the second audio file, playback of the third  
24 audio file would be handled in the same manner as the second audio file selected  
25 above. This process continues until all selected audio files have been played.

1 Although Figs. 4A and 4B show the fading-out of a first audio file and the  
2 fading-in of a second audio file as sequential operations, the fading-out of one  
3 audio file and the fading-in of another audio file are typically performed  
4 simultaneously. For example, block 412 (Fig. 4A) may be performed in parallel  
5 with blocks 414 and 416 (Fig. 4B). These fade-in and fade-out operations may be  
6 performed by cross-fader 118 (Fig. 1).

7 Fig. 5 is a graph illustrating an effective start position and an effective end  
8 position associated with a particular audio file. The audio file is identified by  
9 reference designator 502 and begins at Time = 0 and ends at Time = T. Broken  
10 line 504 represents a threshold amplitude that is used to determine the effective  
11 start position and the effective end position of the audio file. Using this threshold  
12 amplitude eliminates playback of portions of the audio file that are silent or nearly  
13 silent. As shown in Fig. 5, the effective start position of the audio file occurs when  
14 the audio file amplitude crosses threshold 504 and the effective end position of the  
15 audio file occurs when the audio file amplitude again crosses threshold 504.

16 In another embodiment (such as the embodiment discussed above with  
17 respect to Fig. 2 and Fig. 3), a running average is calculated using audio file data  
18 samples. When the running average crosses threshold 504, the effective start  
19 position (or the effective end position) is identified.

20 In another embodiment, the effective start position (or the effective end  
21 position) is identified when a predetermined number of consecutive audio file data  
22 samples exceed (or fall below) the threshold value. Alternatively, the effective  
23 start position (or the effective end position) can be identified when the audio file  
24 amplitude stays above a threshold value for a predetermined amount of time (e.g.,  
25 one second) or falls below the threshold value for a predetermined amount of time.

1 In Fig. 5, the same threshold value 504 is used to determine the effective  
2 start position and the effective end position. In alternate embodiments, different  
3 threshold values may be used to determine the effective start position and the  
4 effective end position of the audio file.

5 Fig. 6 is a graph illustrating the manner in which a first audio file fades-out  
6 and a second audio file fades-in. The first audio file is identified by reference  
7 number 602 and the second audio file is shown as a broken line and is identified  
8 by reference number 604. In Fig. 6, Time = 0 represents the effective start  
9 position of the first audio file. The time between Time = 0 and Time = T1  
10 represents the fade-in period of the first audio file. Time T2 is the effective start  
11 position of the second audio file. Time T3 is the effective end position of the first  
12 audio file. The time between Time = T1 and Time = T2 represents the fade-out  
13 period of the first audio file as well as the fade-in period of the second audio file.

14 In a particular embodiment, a user may create (e.g., "burn") an audio CD  
15 with one or more audio tracks. The cross-fade system attempts to retrieve fade-in  
16 and fade-out parameters from the media library and from the audio files to be  
17 recorded on the CD. If these parameters are not available for a particular audio  
18 file, the cross-fade system scans the audio file and calculates these parameters.  
19 The cross-fade system then scans the audio file a second time to copy the audio  
20 file to the CD while applying the parameters calculated during the first scan of the  
21 audio file. The resulting audio CD contains one or more audio files that include  
22 the necessary fade-in and fade-out parameters.

23 In another embodiment, a user may download one or more audio files from  
24 an online source, such as an online music download service. As the audio files are  
25 downloaded, the cross-fade system attempts to retrieve fade-in and fade-out

1 parameters from the media library and from the audio files being downloaded. If  
2 these parameters are not available for a particular audio file, the cross-fade system  
3 scans the audio file and calculates the parameters as the file is downloaded. The  
4 cross-fade system then saves the fade-in and fade-out parameters in the media  
5 library and/or in the downloaded audio file.

6 The systems and methods discussed herein calculate fade-in and fade-out  
7 parameters at various times, such as during CD ripping, media scans, and the first  
8 time a particular media file is played by the computer system. These parameters  
9 are applied in real-time during playback of the audio data. In many cases, the  
10 fade-in and fade-out parameters are stored in or stored with the associated audio  
11 file. Therefore, if the audio file is copied to a new computer system, the  
12 parameters are readily available to the new computer system without requiring any  
13 additional analysis or calculations. However, certain audio files are read-only  
14 and/or are stored on a read-only media (such as a CD-ROM disc). In this  
15 situation, the fade-in and fade-out parameters associated with the audio files are  
16 available from the media library.

17 Fig. 7 illustrates a general computer environment 700, which can be used to  
18 implement the techniques described herein. The computer environment 700 is  
19 only one example of a computing environment and is not intended to suggest any  
20 limitation as to the scope of use or functionality of the computer and network  
21 architectures. Neither should the computer environment 700 be interpreted as  
22 having any dependency or requirement relating to any one or combination of  
23 components illustrated in the example computer environment 700.

24 Computer environment 700 includes a general-purpose computing device in  
25 the form of a computer 702. One or more media player applications can be



1 executed by computer 702. The components of computer 702 can include, but are  
2 not limited to, one or more processors or processing units 704 (optionally  
3 including a cryptographic processor or co-processor), a system memory 706, and a  
4 system bus 708 that couples various system components including the processor  
5 704 to the system memory 706.

6 The system bus 708 represents one or more of any of several types of bus  
7 structures, including a memory bus or memory controller, a peripheral bus, an  
8 accelerated graphics port, and a processor or local bus using any of a variety of  
9 bus architectures. By way of example, such architectures can include an Industry  
10 Standard Architecture (ISA) bus, a Micro Channel Architecture (MCA) bus, an  
11 Enhanced ISA (EISA) bus, a Video Electronics Standards Association (VESA)  
12 local bus, and a Peripheral Component Interconnects (PCI) bus also known as a  
13 Mezzanine bus.

14 Computer 702 typically includes a variety of computer readable media.  
15 Such media can be any available media that is accessible by computer 702 and  
16 includes both volatile and non-volatile media, removable and non-removable  
17 media.

18 The system memory 706 includes computer readable media in the form of  
19 volatile memory, such as random access memory (RAM) 710, and/or non-volatile  
20 memory, such as read only memory (ROM) 712. A basic input/output system  
21 (BIOS) 714, containing the basic routines that help to transfer information  
22 between elements within computer 702, such as during start-up, is stored in ROM  
23 712. RAM 710 typically contains data and/or program modules that are  
24 immediately accessible to and/or presently operated on by the processing unit 704.  
25

1 Computer 702 may also include other removable/non-removable,  
2 volatile/non-volatile computer storage media. By way of example, Fig. 7  
3 illustrates a hard disk drive 716 for reading from and writing to a non-removable,  
4 non-volatile magnetic media (not shown), a magnetic disk drive 718 for reading  
5 from and writing to a removable, non-volatile magnetic disk 720 (e.g., a "floppy  
6 disk"), and an optical disk drive 722 for reading from and/or writing to a  
7 removable, non-volatile optical disk 724 such as a CD-ROM, DVD-ROM, or other  
8 optical media. The hard disk drive 716, magnetic disk drive 718, and optical disk  
9 drive 722 are each connected to the system bus 708 by one or more data media  
10 interfaces 726. Alternatively, the hard disk drive 716, magnetic disk drive 718,  
11 and optical disk drive 722 can be connected to the system bus 708 by one or more  
12 interfaces (not shown).

13 The disk drives and their associated computer-readable media provide non-  
14 volatile storage of computer readable instructions, data structures, program  
15 modules, and other data for computer 702. Although the example illustrates a hard  
16 disk 716, a removable magnetic disk 720, and a removable optical disk 724, it is to  
17 be appreciated that other types of computer readable media which can store data  
18 that is accessible by a computer, such as magnetic cassettes or other magnetic  
19 storage devices, flash memory cards, CD-ROM, digital versatile disks (DVD) or  
20 other optical storage, random access memories (RAM), read only memories  
21 (ROM), electrically erasable programmable read-only memory (EEPROM), and  
22 the like, can also be utilized to implement the example computing system and  
23 environment.

24 Any number of program modules can be stored on the hard disk 716,  
25 magnetic disk 720, optical disk 724, ROM 712, and/or RAM 710, including by

1 way of example, an operating system 726, one or more application programs 728,  
2 other program modules 730, and program data 732. Each of such operating  
3 system 726, one or more application programs 728, other program modules 730,  
4 and program data 732 (or some combination thereof) may implement all or part of  
5 the resident components that support the distributed file system.

6 A user can enter commands and information into computer 702 via input  
7 devices such as a keyboard 734 and a pointing device 736 (e.g., a “mouse”).  
8 Other input devices 738 (not shown specifically) may include a microphone,  
9 joystick, game pad, satellite dish, serial port, scanner, and/or the like. These and  
10 other input devices are connected to the processing unit 704 via input/output  
11 interfaces 740 that are coupled to the system bus 708, but may be connected by  
12 other interface and bus structures, such as a parallel port, game port, or a universal  
13 serial bus (USB).

14 A monitor 742 or other type of display device can also be connected to the  
15 system bus 708 via an interface, such as a video adapter 744. In addition to the  
16 monitor 742, other output peripheral devices can include components such as  
17 speakers (not shown) and a printer 746 which can be connected to computer 702  
18 via the input/output interfaces 740.

19 Computer 702 can operate in a networked environment using logical  
20 connections to one or more remote computers, such as a remote computing device  
21 748. By way of example, the remote computing device 748 can be a personal  
22 computer, portable computer, a server, a router, a network computer, a peer device  
23 or other common network node, game console, and the like. The remote  
24 computing device 748 is illustrated as a portable computer that can include many  
25 or all of the elements and features described herein relative to computer 702.

1 Logical connections between computer 702 and the remote computer 748  
2 are depicted as a local area network (LAN) 750 and a general wide area network  
3 (WAN) 752. Such networking environments are commonplace in offices,  
4 enterprise-wide computer networks, intranets, and the Internet.

5 When implemented in a LAN networking environment, the computer 702 is  
6 connected to a local network 750 via a network interface or adapter 754. When  
7 implemented in a WAN networking environment, the computer 702 typically  
8 includes a modem 756 or other means for establishing communications over the  
9 wide network 752. The modem 756, which can be internal or external to computer  
10 702, can be connected to the system bus 708 via the input/output interfaces 740 or  
11 other appropriate mechanisms. It is to be appreciated that the illustrated network  
12 connections are exemplary and that other means of establishing communication  
13 link(s) between the computers 702 and 748 can be employed.

14 In a networked environment, such as that illustrated with computing  
15 environment 700, program modules depicted relative to the computer 702, or  
16 portions thereof, may be stored in a remote memory storage device. By way of  
17 example, remote application programs 758 reside on a memory device of remote  
18 computer 748. For purposes of illustration, application programs and other  
19 executable program components such as the operating system are illustrated herein  
20 as discrete blocks, although it is recognized that such programs and components  
21 reside at various times in different storage components of the computing device  
22 702, and are executed by the data processor(s) of the computer.

23 Various modules and techniques may be described herein in the general  
24 context of computer-executable instructions, such as program modules, executed  
25 by one or more computers or other devices. Generally, program modules include

1 routines, programs, objects, components, data structures, etc. that perform  
2 particular tasks or implement particular abstract data types. Typically, the  
3 functionality of the program modules may be combined or distributed as desired in  
4 various embodiments.

5 An implementation of these modules and techniques may be stored on or  
6 transmitted across some form of computer readable media. Computer readable  
7 media can be any available media that can be accessed by a computer. By way of  
8 example, and not limitation, computer readable media may comprise "computer  
9 storage media" and "communications media."

10 "Computer storage media" includes volatile and non-volatile, removable  
11 and non-removable media implemented in any method or technology for storage  
12 of information such as computer readable instructions, data structures, program  
13 modules, or other data. Computer storage media includes, but is not limited to,  
14 RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM,  
15 digital versatile disks (DVD) or other optical storage, magnetic cassettes, magnetic  
16 tape, magnetic disk storage or other magnetic storage devices, or any other  
17 medium which can be used to store the desired information and which can be  
18 accessed by a computer.

19 "Communication media" typically embodies computer readable  
20 instructions, data structures, program modules, or other data in a modulated data  
21 signal, such as carrier wave or other transport mechanism. Communication media  
22 also includes any information delivery media. The term "modulated data signal"  
23 means a signal that has one or more of its characteristics set or changed in such a  
24 manner as to encode information in the signal. By way of example, and not  
25 limitation, communication media includes wired media such as a wired network or

1 direct-wired connection, and wireless media such as acoustic, RF, infrared, and  
2 other wireless media. Combinations of any of the above are also included within  
3 the scope of computer readable media.

4 Although the description above uses language that is specific to structural  
5 features and/or methodological acts, it is to be understood that the invention  
6 defined in the appended claims is not limited to the specific features or acts  
7 described. Rather, the specific features and acts are disclosed as exemplary forms  
8 of implementing the invention.